

Homework # 3

ENAE 441

Instructions

Please answer the following questions. Your answers should be uploaded to gradescope in two files:

1. `{your_name}.pdf` — derivations and / or written explanations of results
2. `submission.py` — the python script for the autograder (must be named `submission.py`!)

Note the type of question is indicated by its color:

- **Blue** questions are solved via pen and paper and uploaded in your **PDF file**.
- **Red** questions correspond to **autograded functions** that must be populated in `submission.py`.
- **Black** questions solutions are to be answered in `submission.py` but are **not autograded** (e.g. plots, discussions, etc)

Programming Questions

Each assignment has starter code available on ELMS. Your grade depends exclusively on the output of functions labeled `#REQUIRED`. Additional helper functions are included in the starter code to assist with debugging.

Question 1. (30 pts) Propagation Functions

Please implement the following functions in your code to confirm your understanding of the different propagation strategies. You are not required to use these exact functions in subsequent questions, as there may be more appropriate / efficient implementations for subsequent questions. They are merely provided as a way to confirm your own understanding of the process.

- a) Write a function that propagates a LTI system forward numerically ¹ and plot the displacement $x(t)$ and velocity $\dot{x}(t)$.
- b) Write a function that propagates a LTI system forward analytically using a continuous time (CT) STM ².
- c) Write a function that propagates a LTI system forward analytically using a discrete time (DT) STM ³.
- d) Write a function that propagates a LTV system by jointly integrates the nominal trajectory and the state transition matrix for a fixed amount of time. ⁴.
- e) Write a function that returns the maximum allowable sampling time for a discrete time system to avoid aliasing.

¹using Python's `scipy.integrate.solve_ivp`

²Be sure to use `scipy.linalg.expm` to take the matrix exponential

³Be sure to use `numpy.linalg.matrix_power` to raise the matrix to a power

⁴Hint: Form a joint state-STM vector $\mathbf{z} = [\mathbf{x}_{\text{nom}}(t) \quad \phi(t, t_0)]^\top$. Unroll the STM into a single column vector such that the object passed to solve ivp is a single column vector. Use $\Phi(t_0, t_0) = \mathcal{I}_{6 \times 6}$ for the initial condition

Question 2. (30 pts) Continuous Time Linear System

Consider a spring-mass-damper system

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0$$

with system mass $m = 1$, damping coefficient $c = 0.5$ Ns/m, and spring constant $k = 4$ N/m and an initial condition of $\mathbf{x}(0) = [1, 0]^T$.

The system is measured by observing the position

$$y(t) = x(t)$$

- a) Express the system as a continuous-time state-space model **and** as a discrete-time model. Define a state vector of $\mathbf{x}(t) = [x(t), \dot{x}(t)]^T$ and define all of the corresponding matrices. Assume no external input ($\mathbf{u}(t) = 0$).
- b) Propagate the system forward for 10 seconds using all three LTI propagation methods and plot $x(t)$ and $\dot{x}(t)$ on the same figure.
- c) Compare the results. What happens when you apply a Δt value that exceeds the critical threshold in the DT STM?
- d) Use the set of position measurements in **HW3-spring-data.npy** to determine the initial state $\mathbf{X}(t = 0)$ for a different trajectory.
- e) How many measurements are needed to ensure the state $\mathbf{x}(t)$ is observable if $\Delta t = 1$ s? Explain your findings.

Question 3. (40 pts) Continuous Time Non-Linear System

Consider a satellite in a 3D Keplerian orbit around Earth. The dynamics of the satellite can be modeled using Newton's second law under the gravitational force:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r},$$

where $\mu = 398600 \text{ km}^3/\text{s}^2$ is the Earth's gravitational parameter and \mathbf{r} is the position vector of the satellite in the Earth-centered inertial (ECI) frame.

The initial position and velocity vectors are:

$${}^{\mathcal{N}}\mathbf{X}(0) = \begin{bmatrix} 7000 \text{ km} \\ 0.0 \text{ km} \\ 0.0 \text{ km} \\ 0.0 \text{ km s}^{-1} \\ 7.5 \text{ km s}^{-1} \\ 3.5 \text{ km s}^{-1} \end{bmatrix}^{\top}$$

and the measurement function of the system is the range, ρ taken from an observer at $\phi = 30^\circ$ latitude and $\lambda = 60^\circ$ longitude, where the radius of the Earth, R_E is taken as 6378km and the rotation of the Earth is $\omega_{E/N} = 7.2921150 \times 10^{-5} \text{ rad/s}$. Define the state vector as $\mathbf{x}(t) = [\mathbf{r}(t), \mathbf{v}(t)]^{\top}$.

- a) Linearize the system around a nominal trajectory $X_{nom}(t)$, deriving the system and output matrices $A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ and $C = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$.
- b) Write a function that outputs A_k .
- c) Write a function that outputs C_k .
- d) Numerically propagate (i) the nominal trajectory and (ii) the nominal trajectory perturbed an initial state deviation of $\delta\mathbf{X} = [30, 0, 0, 0, 0, 0.1]^{\top}$ for 90 minutes. Plot the difference in the position vector $\delta\mathbf{r}(t)$ as a function of time.
- e) Propagate the state deviation directly using the integrated state transition matrix from the augmented nominal state vector (i.e. problem 1d) and plot the position vector deviation as a function of time.
- f) Increase the initial state perturbation to $\delta\mathbf{X} = [1000, 0, 0, 0, 0, 0.1]^{\top}$ and integrate (1) numerically, and (2) using the STM. Plot $\delta\mathbf{r}(t)$ for both propagation strategies.
- g) Explain what you see.
- h) A set of range measurements are provided for a different trajectory in `HW3-kepler-data.npy` from the same observer. Use these measurements to estimate the initial state deviation $\delta\mathbf{x}(t_0)$ using the same nominal trajectory as before.
- i) Explain your approach.